BAB 02

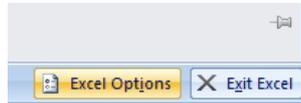
Sebelum Membuat Fungsi dan Formula Sendiri, Kenalilah Macro

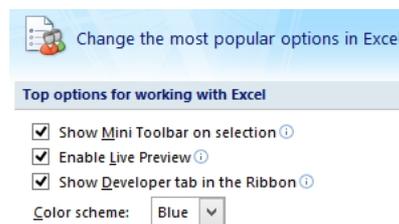
Ujung tombak dalam pembuatan fungsi terletak di dalam Macro, yaitu fitur paling utama kalau Anda sebagai pengguna MS Excel menyukai pembuatan baris demi baris script pemrograman. Fitur ini memang bukan untuk semua kalangan. Hanya orang-orang tertentu yang memanfaatkan Macro dengan begitu intensif. Salah satunya adalah mereka yang suka menciptakan fungsi-fungsi.

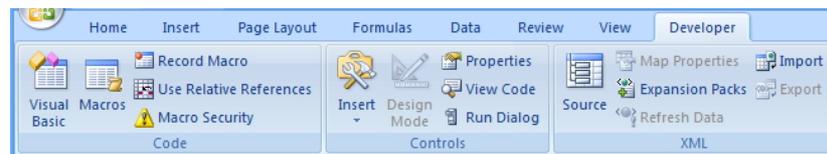
Bab ini membantu Anda mengenali Macro dengan baik. Jika Anda telah mengenal Macro beserta seluk-beluknya, maka bab ini tidak perlu Anda pelajari. Di dalam bab ini, Anda akan mengenali Macro dan hubungannya dengan MS Excel.

Macro di dalam MS Excel

Ada fitur khusus di dalam MS Excel yang harus "diaktifkan" terlebih dulu agar Anda bisa bekerja dengan menggunakan Macro. Fitur itu diletakkan di dalam Tab Developer. Untuk memunculkan Tab Developer, maka lakukanlah langkah-langkah sebagai berikut:

1. Tekanlah tombol **Office** dan klik tombol **Excel Options**.

Memilih tombol Excel Options

2. Aktifkan kotak cek **Show Developer Tab in the Ribbon**.

Mengaktifkan kotak cek untuk Developer Tab

3. Tekanlah tombol **OK**. Kini akan terlihat Tab Developer di sisi paling kanan dari jajaran Tab tersebut.

Tab Developer yang paling bertanggung jawab untuk pembuatan Macro

Ketika Anda mengklik Tab Developer, maka akan Anda temukan berbagai kontrol untuk membantu pembuatan Macro memakai Visual Basic for Application.

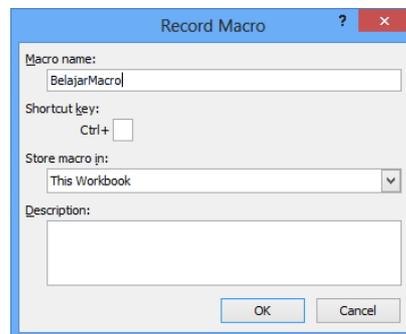
Cara Membuat Macro Paling Mudah

Saat Tab Developer sudah aktif, maka Anda sudah bisa membuat Macro. Cara termudah membuat Macro adalah dengan menggunakan fitur perekaman. Dalam praktiknya, pembuatan Macro menggunakan fitur perekaman mungkin tidak bisa dijadikan pedoman pembuatan fungsi. Namun demikian, Anda perlu memahami cara pembuatan Macro memakai fitur perekaman ini agar wawasan Anda tentang Macro menjadi lebih lengkap.

Berikut langkah-langkah untuk membuat Macro menggunakan fitur perekaman:

1. Buatlah workbook baru dengan menekan tombol **Ctrl+N** agar Anda bisa mulai mengenal Macro benar-benar dari nol. Pastikan Anda sudah mengaktifkan Tab **Developer**.
2. Klik ikon **Use Relative References** yang ada di dalam kelompok **Code**.

Ikon Use Relative References yang ada di dalam area Code

3. Pilih salah satu sel di dalam worksheet, misalnya sel A1.
4. Klik **Record Macro**. Setelah muncul jendela **Record Macro**, buatlah nama yang spesifik di dalam kotak teks **Macro Name**, misalnya: ***BelajarMacro***.

Membuat nama baru untuk proyek Macro ini

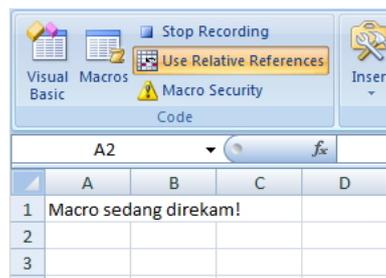
5. Pada kotak teks **Ctrl+**, tekan tombol **Shift+M**. Atau Anda bisa menekan kombinasi tombol yang lain. Pada intinya, untuk menjalankan rekaman Macro ini nanti, Anda harus menekan tombol **Ctrl+Shift+M**.
6. Pada menu kombo **Store Macro In** biarkan opsi yang aktif tetap **This Workbook**. Dengan demikian, rekaman Macro ini bisa dijalankan di seluruh worksheet pada satu workbook.

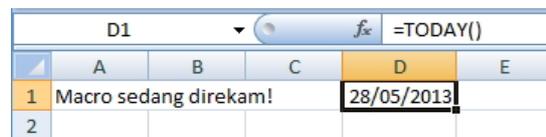
7. Pada kotak teks **Description**, Anda bisa menulis deskripsi atas macro ini nantinya.
8. Tekanlah tombol **OK.**

Saat ini Anda masuk atau berada di dalam mode **Record Macro**. Jadi, apa pun yang akan Anda lakukan nanti setelah menekan tombol OK di atas akan "direkam" oleh Macro.

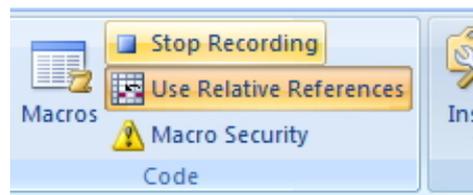
Proses Merekam Macro

Selama di dalam mode Record Macro, Anda bisa mengerjakan tugas-tugas tertentu yang nanti bisa diulang-ulang terus sesuai kebutuhan. Berikut contoh aktivitas yang bisa direkam menggunakan Macro:

1. Mulailah dengan mengetik teks di dalam sel A1.

Menulis teks di dalam sel yang aktif

2. Klik sel D1.
3. Di dalam sel tersebut, ketik fungsi sederhana berikut ini: ***=TODAY()***.

Memasukkan tanggal dengan memanfaatkan formula

4. Klik lagi sel D1 tempat fungsi itu ditulis. Lantas, klik-kanan dan pilih ikon untuk mengubah warna teks menjadi merah.

5. Kembalilah lagi ke dalam Tab **Developer** dan kali ini, klik ikon **Stop Recording**.

Pilih Stop Recording untuk menghentikan aksi perekaman

Pada langkah-langkah di atas, Anda sudah berhasil merekam Macro. Selanjutnya, kita perlu mengenali bagaimana Macro di atas akan memengaruhi cara kita bekerja.

Menjalankan Macro Hasil Perekaman

Macro yang telah selesai direkam itu kini bisa "diputar ulang". Untuk mengecek Macro yang tadi telah Anda buat, lakukanlah langkah-langkah di bawah ini:

1. Klik sel lain manapun yang Anda inginkan.
2. Tekan tombol shortcut yang telah Anda atur tadi (tombol **Ctrl+Shift+M**).
3. Anda akan melihat tulisan seperti yang tadi telah dibuat.

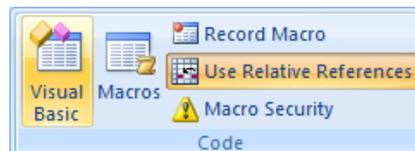
Teks yang berhasil dibuat sama persis dengan teks semula

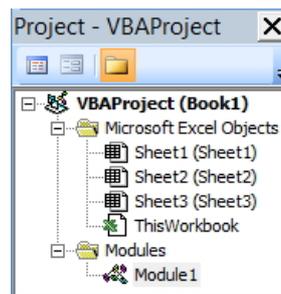
Anda akan melihat teks yang sama persis dengan teks dan formula yang tadi telah dibuat.

Melihat Struktur Macro Hasil Recording

Macro disusun menggunakan VBA, atau *Visual Basic for Application*. VBA adalah script pemrograman yang dimanfaatkan untuk menciptakan aplikasi yang dikenal dengan sebutan Macro. Jadi selama proses recording berjalan, maka 'di balik layar' MS Excel akan menyusun baris-baris script VBA.

Untuk melihat struktur VBA itu, Anda bisa melihat langkah-langkah sebagai berikut:

1. Pada Tab **Developer**, klik ikon **Visual Basic** yang ada di dalam kelompok **Code.**

Memilih ikon Visual Basic

2. Nanti Anda akan melihat jendela **Microsoft Visual Basic**.
3. Klik ikon + yang ada di samping kanan **VBAProject (Book1)** pada panel **Project – VBAProject**. Setelah itu, klik tanda + di kiri **Modules.**
4. Klik-ganda **Module1.**

Untuk melihat script, klik-ganda pada Module1

Setelah mengklik-ganda Module1, maka akan terlihat script VBA lengkap yang menyimpan langkah-langkah yang tadi telah kita buat selama proses recording berlangsung.

```
(General)                          BelajarMacro

Sub BelajarMacro()
'
' BelajarMacro Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Macro sedang direkam!"
    ActiveCell.Offset(1, 0).Range("A1").Select
    Application.WindowState = xlNormal
    ActiveCell.Offset(-1, 3).Range("A1").Select
    ActiveCell.FormulaR1C1 = "=TODAY()"
    ActiveCell.Select
    ActiveCell.FormulaR1C1 = "=TODAY()"
    ActiveCell.Select
    With Selection.Font
        .Color = -16776961
        .TintAndShade = 0
    End With
End Sub
```

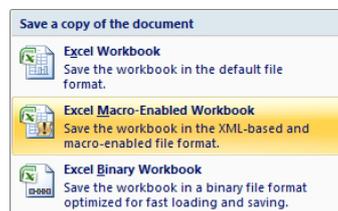
Script yang akan memainkan hasil rekaman Macro yang sudah dibuat tadi

Menyimpan Workbook yang Mengandung Macro

Cara menyimpan workbook ke dalam sebuah file tidaklah sama antara workbook tanpa Macro dan workbook ber-Macro. Jika Anda bekerja dengan workbook yang mengandung Macro, maka workbook tersebut harus disimpan ke dalam format yang mendukung penyimpanan Macro.

Jika Anda bekerja dengan menggunakan MS Excel 2007/2010, maka workbook itu harus disimpan ke dalam format XLSM.

Berikut langkah-langkahnya:

1. Tekan tombol **Office** dan pilihlah **Save As**.
2. Klik pada opsi **Excel Macro-Enabled Workbook**.

Pilih Excel Macro-Enabled Workbook

3. Lakukan penyimpanan file seperti biasanya.

Jika ingin agar workbook itu bisa disimpan dan dibuka untuk MS Excel versi 2003, maka Anda bisa memilih **Excel 97-2003 Workbook**. Macro tetap akan tersimpan bersama-sama dengan workbook namun hanya akan dikenali sebagai workbook MS Excel versi 97-2003.

Mengatasi Macro yang dalam Kondisi Disabled

Jangan anggap enteng Macro. VBA, script yang berada di balik Macro, merupakan script yang sebenarnya sangat "berbahaya". Anda bisa membuat macro sederhana yang bisa menghapus file, merusak workbook, mengganggu MS Windows, atau menghapus data-data yang ada di workbook. Oleh karena itu, tak heran jika sejak diluncurkan pertama kali, MS Excel 2007 dan 2010 telah dilengkapi dengan fitur-fitur pengaman.

Bagaimana cara kerja fitur pengaman itu? Lihat langkah-langkah berikut:

1. Buka file workbook MS Excel 2007/2010 yang mengandung Macro dengan mengklik **Office Button** dan klik **Open**.
2. Lihat gambar di bawah ini. Begitu workbook itu dibuka, Macro tersebut akan dinonaktifkan oleh MS Excel. Anda akan melihat bar yang menunjukkan Macro itu dalam kondisi nonaktif seperti pada gambar di bawah ini.

Macro akan dinonaktifkan secara otomatis lewat pesan peringatan "Security Warning" di atas

3. Lantas, untuk mengaktifkan Macro itu, tekan tombol **Options**.
4. Pilihlah opsi **Enable This Content**. Lanjutkan dengan menekan tombol **OK**.

Aktifkan Macro dengan memilih Enable this content

5. Workbook dan Macro di dalamnya pun akan dibuka dengan normal.

Sebelum melakukan langkah-langkah di atas, pastikan Anda mengetahui persis apa isi Macro yang ada di dalam workbook itu sebelum Anda menyadari betapa berisikonya jika Macro itu tetap diaktifkan.

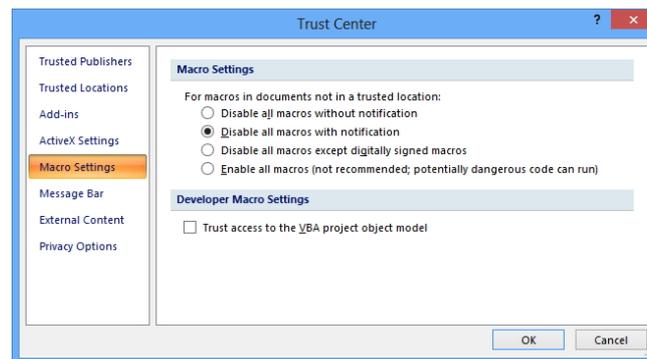
Jika muncul kotak dialog seperti gambar di atas, cukup pilih tombol Enable Macros

Mengatur Keamanan Macro

Jika komputer yang digunakan untuk mengolah workbook ber-Macro hanya digunakan untuk pegawai administratif, maka Anda bisa mempertimbangkan untuk menonaktifkan fitur keamanan Macro itu.

MS Excel 2007/2010 menyediakan fitur untuk mengelola keamanan Macro. Jika Anda membuka Tab Developer, maka akan Anda temukan ikon **Macro Security** yang ada di dalam kelompok **Code**.

Memilih ikon Macro Security

Di situ, Anda akan melihat berbagai pilihan pengaturan keamanan untuk Macro di dalam area **Macro Settings**. Pilih salah satu pilihan sesuai dengan pertimbangan Anda atas risiko yang nanti terjadi, yaitu:

Pengaturan keamanan Macro

- **Disable all macros without notification.** Macro yang ada di dalam sebuah workbook tidak akan dijalankan ketika file workbook itu dibuka dan Anda tidak akan melihat notifikasi apa pun seperti contoh di atas.
- **Disable all macros except digitally signed macros.** Macro dalam sebuah workbook tidak akan dijalankan ketika file workbook itu dibuka, kecuali jika workbook itu dibuat oleh pengembang yang dianggap terpercaya.
- **Disable all macros with notification.** Macro akan dinon-aktifkan namun MS Excel akan menampilkan notifikasi untuk memberi Anda pilihan bilamana Macro itu akan diaktifkan atau tidak.

- **Enable all macros (not recommended; potentially dangerous code can run).** Macro yang ada di dalam sebuah workbook akan selalu diaktifkan secara otomatis. Opsi ini berisiko namun efisien jika Anda menyediakan komputer khusus untuk mengoperasikan Macro tersebut (misal: untuk komputer administrasi).

Dengan adanya pilihan-pilihan di atas, jika Anda ingin membuat fungsi sendiri, maka seluruh pilihan kecuali pilihan nomor satu dan dua, bisa dipilih.

Meluncurkan Visual Basic Editor

Ketika nanti Anda membuat fungsi sendiri menggunakan Macro, lantas di mana fungsi dan Macro itu dibuat? MS Excel menyediakan area khusus yang disebut dengan istilah Visual Basic Editor yang berfungsi sebagai tempat untuk merangkai script pemrograman *Visual Basic for Application* agar bisa menciptakan fungsi sendiri, atau Macro pada umumnya untuk menyelesaikan tugas yang spesifik.

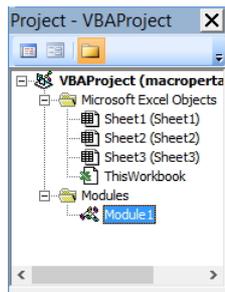
Visual Basic Editor hanya bisa diluncurkan jika terlebih dulu Anda menjalankan MS Excel. Ada beberapa metode meluncurkan Visual Basic Editor. Selain dengan menggunakan ikon **Visual Basic** yang ada di dalam Tab Developer, Anda juga bisa menjalankan Visual Basic Editor dengan menekan tombol **Alt+F11**.

Mengenal Jendela Visual Basic Editor

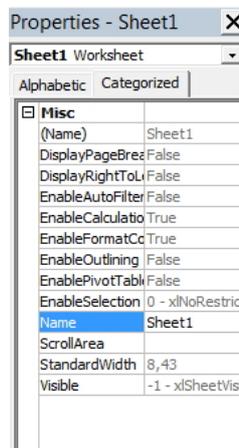
Visual Basic Editor dibagi dalam beberapa panel. Secara sekilas berikut fungsi dari masing-masing panel itu.

Project Explorer

Panel Project Explorer dapat ditemukan di pojok kiri atas. Ada berbagai fungsi untuk panel ini. Namun yang paling utama adalah untuk membuka VBA yang bekerja di masing-masing worksheet. Sebagai contoh, jika ingin membuka VBA yang bekerja hanya untuk Sheet1, maka cukup klik Sheet1 (Sheet1). Sedangkan kalau ingin membuka VBA yang berlaku di seluruh workbook, maka klik ThisWorkbook.

Jika tidak menemukan panel ini, maka Anda bisa mengaktifkannya dengan menekan menu **View > Project Explorer**. Atau, percepat dengan menekan tombol **Ctrl+R** atau klik ikon yang ada di dalam toolbar.

Panel Project Explorer untuk mengeksplorasi objek-objek di dalam proyek Macro

Panel Properties

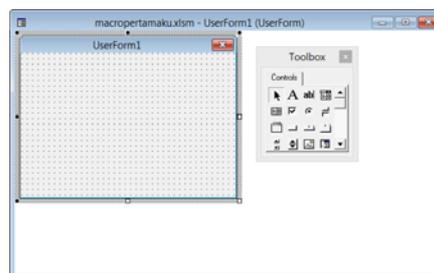
Panel Properties berfungsi untuk menampilkan berbagai macam properti yang melekat pada sebuah objek. Jika Anda belum menemukan panel ini, maka aktifkanlah dengan menekan menu **View > Properties Window** atau cukup dengan menekan tombol **F4.**

Panel Properties akan memiliki tampilan beragam, tergantung objek yang aktif saat itu

Panel Code

Panel Code adalah panel utama di mana Anda menulis script VBA. Inilah panel terbesar yang ada di dalam Visual Basic Editor. Fungsi panel Code adalah untuk mengetik, melihat, merancang, dan melakukan fungsi editing pada script VBA agar Anda bisa menciptakan fungsi-fungsi sendiri.

Panel Code adalah panel dengan ukuran paling besar dan mungkin juga yang berperan paling penting dalam pembuatan VBA

Panel Object

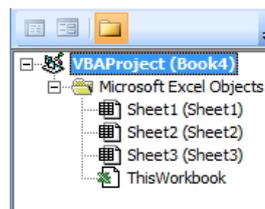
Panel ini umumnya jarang digunakan untuk pembuatan fungsi sendiri. Namun kalau Anda ingin merancang Macro yang user oriented, maka Anda membutuhkan panel Object. Jika panel Object ini belum terlihat di dalam layar monitor, maka munculkan panel ini dengan mengklik menu **View > Object** atau dengan menekan tombol **Shift+F7**.

Tampilan panel Object

Menulis Visual Basic for Application di dalam Panel Code

Macro tak akan pernah lepas dari script VBA. Keduanya tampak menjadi satu. Oleh karena itulah, saat membuat fungsi-fungsi, maka Anda harus memanfaatkan script VBA.

Secara garis besar, script atau kode-kode VBA itu ditulis di dalam panel Code lewat dua metode, yaitu dengan menulis (mengetik) kode langsung secara manual dan atau menggunakan Macro Recorder yang nanti selanjutnya 'berubah wujud' menjadi kode-kode VBA.

Untuk pembuatan fungsi-fungsi, maka secara umum Anda harus menempuh metode kedua, yaitu menulis script secara manual. Berikut salah satu langkah-langkah untuk menulis kode secara langsung dan manual untuk menciptakan Macro sederhana:

1. Buatlah workbook baru dan masuklah ke dalam **Visual Basic Editor** (tekan saja tombol **Alt+F11**).
2. Klik pada nama workbook yang tercatat di dalam panel **Project.**

Klik pada nama workbook di mana Macro akan dibuat

3. Buatlah module baru dengan mengklik menu **Insert > Module**.
4. Kalau sudah, ketik script di bawah ini di dalam panel Code:

```
Sub PilihSalahSatu()
    Dim Jawaban As Integer
    Jawaban = MsgBox("Apakah Macro itu sulit?", vbYesNo)
    Select Case Jawaban
        Case vbYes
```

```
            MsgBox "Baca dulu donk buku ini...!"
        Case vbNo
            MsgBox "Perdalam ilmu Macro Anda dari buku ini!"
        End Select
End Sub
```

5. Tekan tombol **F5** untuk menjalankan script VBA di atas. Di layar monitor, Anda akan melihat kotak dialog dengan pertanyaan di bawah ini.

Kotak dialog lengkap dengan dua alternatif jawaban

6. Jika menjawab **Yes** atau **No**, maka akan muncul kotak dialog dengan jawaban-jawaban yang berbeda. Lihat contohnya di bawah ini.

Salah satu jawaban akan muncul di dalam kotak dialog

Di atas tertulis contoh script Visual Basic Application yang melibatkan kotak dialog. Anda bisa memodifikasi script di atas jika sudah mengenal hakekat pemrograman VBA.

Bekerja dengan Menggunakan Object

Saat Anda bekerja dengan menggunakan *Visual Basic for Application* (VBA) untuk tugas apa pun, maka Anda perlu melihat MS Excel sebagai

objek satu-utuh yang di dalamnya terdiri atas objek-objek lain yang lebih kecil. Oleh karena itu, Anda akan bekerja dengan objek yang di dalamnya terdapat objek lain yang ada di bawahnya. Jadi, objek-objek itu menjadi terstruktur secara hierarki.

Mengenal Hierarki dan Collection Sebuah Objek

MS Excel adalah sebuah objek besar yang utuh dan diberi nama ***Application Object***. Di dalam object ini, terdapat objek-objek lainnya yang lebih kecil sehingga membentuk struktur hierarki. Sebagai contoh, Application Object mengandung objek-objek yang lebih kecil, seperti: ***AddIn***, ***Window***, ***Workbook***, ***WorksheetFunction***.

Nah, di dalam objek-objek di atas kita bisa urai lagi objek-objek yang lebih kecil. Sebagai contoh, objek Workbook terdiri atas beberapa objek yang lebih kecil sebagai berikut: ***Chart, Name, VBProject, Window, Worksheet***.

Nanti, di dalam objek Worksheet di atas, kita juga akan temukan objek-objek lainnya, misalnya: ***Comment***, ***Hyperlink***, ***Name***, ***PageSetup***.

Collections (koleksi) adalah kumpulan objek yang memiliki tipe yang sama. Sebuah koleksi ditulis dengan penulisan huruf "s" (berarti jamak dalam Bahasa Inggris) di belakang penamaan koleksi tersebut. Contohnya sebagai berikut: ***Workbooks*** (koleksi-koleksi yang berisi objek-objek Workbook yang saat itu sedang aktif), ***Worksheets*** (collection yang berisi objek-objek worksheet yang ada di dalam sebuah objek Workbook), ***Charts*** (collection dari objek Chart yang ada di dalam sebuah objek Workbook), ***Sheets*** (collection dari objek Worksheet yang ada di dalam sebuah objek Workbook).

Ketika Anda ingin menggunakan sebuah object, atau collection, maka Anda harus mengidentifikasi keduanya itu dengan cara menuliskannya secara tepat.

Dalam penulisannya, objek dan collection dipisahkan menggunakan tanda titik ".". Sebagai contoh, jika Anda ingin memilih Sheet1 dalam sebuah workbook, maka cukup tuliskan:

```
Sheets("Sheet1").Select
```

Jika ingin merujuk pada workbook yang ditunjuk pada file XLS tertentu, maka nama workbook itu harus disebut di dalam collections Workbooks. Misalnya seperti ini

```
Workbooks("Datapenjualan.xls").Sheets("Sheet1").Select
```

Ingat, beri tanda kutip ganda untuk merujuk pada objek. Jika tidak menggunakan tanda kutip ganda maka tulisan tersebut, misalnya Sheet1 (tanpa tanda kutip), akan dianggap sebagai variable.

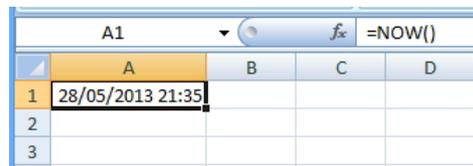
Bekerja dengan Menggunakan Property

Property, atau biasa ditulis Properti, adalah karekteristik yang dimiliki atau melekat pada sebuah objek. Dalam praktiknya, properti ini akan membuat object yang telah kita tunjuk di atas menjadi lebih fungsional.

Sebagai contoh, jika Anda menunjuk objek worksheet "Sheet1" maka Anda hanya sekadar menunjuk saja. Selanjutnya, Anda bisa melakukan sesuatu terhadap Sheet1 itu, misalnya dengan menulis data di dalamnya.

```
Workbooks("Datapenjualan.xlsx").Worksheet("Sheet1")
```

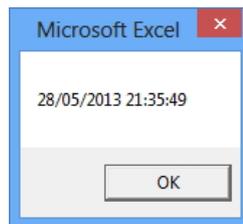
Maka itu, kita pilih gunakan properti. Dalam contoh di atas, properti tersebut adalah Value. Untuk menguji bagaimana properti di atas bekerja, maka lakukanlah langkah-langkah sebagai berikut:

1. Buatlah workbook baru.
2. Di dalam sel A1 tulislah function: ***=NOW()***.

Tulislah data di dalam sel A1

3. Masuklah ke dalam Visual Basic Editor.
4. Tulis script seperti berikut ini di dalam panel Code:

```
Sub Tampilkan()
   DataDitampilkan = Worksheets("Sheet1").Range("A1").Value
   MsgBox DataDitampilkan
End Sub
```

5. Setelah itu, tekanlah tombol **F5**.

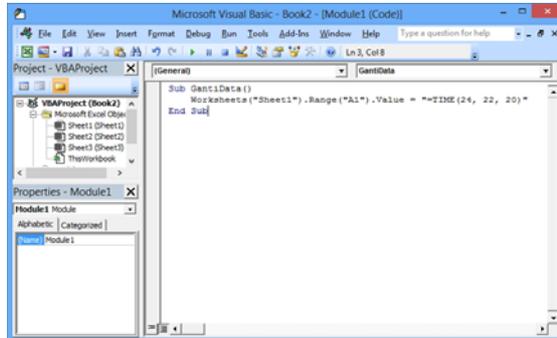
Nanti, Anda akan melihat teks yang sama seperti teks yang tertulis di dalam sel A1. Teks tersebut, yang berbentuk tanggal, didapat dengan mengambil properti dari object bernama Range.

Angka di dalam kotak dialog

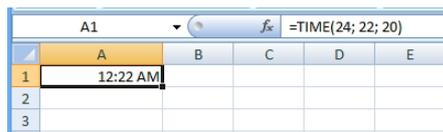
Mengubah Property Sebuah Objects

Kalau nilai di dalam sebuah property bisa ditampilkan, maka itu artinya nilai tersebut juga bisa diubah menggunakan VBA. Dengan kata lain, nilai di dalam sebuah sel yang ada pada worksheet bisa dikontrol lewat VBA. Ada beberapa cara untuk mengubah nilai property sebuah object.

Menulis Value Langsung Menggunakan VBA

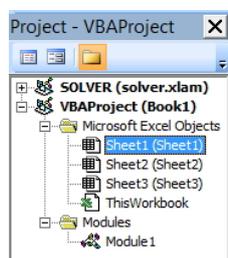
Untuk mengubah nilai property di dalam sebuah object, Anda bisa menulis nilai yang baru itu secara langsung lewat script. Jadi, Anda mengontrol apa yang terlihat di dalam worksheet lewat VBA. Contohnya sebagai berikut:

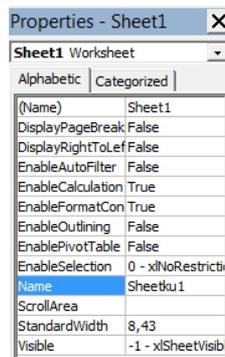
```
Sub GantiData()
  Worksheets("Sheet1").Range("A1").Value = "=TIME(24, 22, 20)"
End Sub
```

Script lanjutan untuk mengubah nilai di dalam object Range("A1")

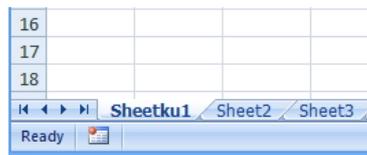
Yang terjadi kemudian, di dalam sel A1 akan tertulis nilai dari fungsi Time, yaitu: 24:22:20.

Angka di dalam sel A1 yang telah berubah

Pengubahan Property Menggunakan Panel Properties

Walaupun tidak semua, ada beberapa property yang nilainya bisa diubah tanpa harus melewati penulisan script secara langsung. Cara pengubahannya dengan memanfaatkan panel **Properties**. Sebagai contoh, lakukan langkah-langkah sebagai berikut:

1. Klik pada **Sheet1(Sheet1)** yang ada di dalam panel **Project**.

Memilih Sheet1 di dalam panel Project

2. Di dalam panel **Properties**, ubah value untuk **Name** menjadi nama yang lain, misalnya: ***Sheetku1***.

Mengubah value Name menjadi Sheetku1

3. Beralihlah ke dalam worksheet MS Excel dan lihat nama sheet yang sudah berganti itu.

Nama Sheet1 yang sudah berubah menjadi Sheetku1

Umumnya, yang bisa diganti menggunakan panel Properties antara lain: Name, Height, Width, Left, Top, dan lainnya.

Bekerja dengan Menggunakan Methods

Methods berkaitan dengan masalah apa yang Object itu bisa lakukan. Oleh karena itu, maka secara langsung method bisa memengaruhi nilai sebuah property.

Lihatlah contoh script di bawah ini:

```
Sub HapusData()
    Range("A1").ClearContents
End Sub
```

Jika script di atas dijalankan, maka sel A1 akan dibersihkan dari data apa pun yang ada di dalamnya. Sebuah methods bisa memiliki arguments. yang akan mengatur apa yang akan dilakukan oleh method itu terhadap objek. Untuk menulis arguments, Anda perlu memisahkan arguments itu dengan method memanfaatkan tanda spasi.

Mari kita bedah script di bawah ini:

```
Sub KopiData()
    Worksheets("Sheet1").Activate
    Range("A1").Copy Range("B1")
End Sub
```

Pada script di atas, akan terlihat adanya dua buah method, yaitu Activate untuk memilih Sheet1 pada sebuah worksheet lewat pernyataan object Worksheets("Sheet1") dan Copy. Kita menggunakan method Copy untuk mengopi data yang ada di dalam object Range("A1"), yaitu data di dalam sel A1. Selanjutnya, method Copy ini memiliki argument yang dinyatakan dalam bentuk tujuan di mana data yang telah dikopi itu akan diletakkan. Dalam hal ini, argument-nya adalah Range("B1").

Mengenal Procedure dan Function

Pokok dan inti bahasan buku ini sebenarnya bermuara pada Procedure dan Function. Itu artinya, ketika Anda ingin menciptakan fungsi-fungsi buatan sendiri untuk menyelesaikan sebuah kasus, maka Anda akan sangat sering bekerja dengan menggunakan Procedure dan Function ini.

Dalam dunia pemrograman istilah **sub procedure** dan **function** merupakan istilah "baku". Artinya, kedua istilah itu pasti akan Anda temukan. Pertanyaannya, bagaimana membedakan antara Sub Procedure dan Function?

Sub Procedure

Sub procedure dapat diartikan sebagai kumpulan kode script VBA yang digunakan untuk melakukan aksi tertentu di dalam MS Excel. Dengan demikian, sub procedure lebih menyerupai sebuah perintah dan ketika perintah itu dijalankan, maka akan terjadi sesuatu seperti yang kita harapkan. Sebagai contoh, dengan memanfaatkan procedure kita bisa

membuat kode untuk mewarnai latar belakang sel menjadi warna tertentu. Ketika kode itu dijalankan, maka warna latar belakang sel benar-benar berubah menjadi warna tertentu.

FUNCTION

Sedangkan function dapat diartikan sebagai kumpulan kode script VBA yang berfungsi untuk melakukan kalkulasi atau penghitungan. Jadi di dalam function biasanya terdapat rumus-rumus yang kita ciptakan sendiri. Dengan demikian, ketika function dijalankan maka kita akan melihat nilai tertentu di dalam MS Excel.

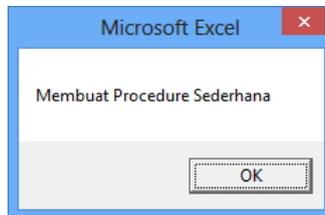
Sebenarnya, kata "function" sendiri tidak asing di dalam dunia MS Excel. Kalau kita ingin menjumlah dengan menggunakan function, maka kita bisa menggunakan function: =SUM(). Function =SUM() ini tersusun dari kode-kode rumus yang berfungsi untuk mengolah argument-argument, biasanya data-data sel, agar angka-angka di dalam sel itu bisa dijumlahkan. Nah dalam buku ini, Anda akan sangat sering berurusan dengan Function.

Cara Membuat Sub Procedure

Membuat sub procedure pada dasarnya gampang dan sederhana. Awali dengan penulisan: **Sub** dan tekan tombol **Enter**. Secara otomatis Visual Basic Editor akan menambahkan dengan script: **End Sub** di akhir script itu. Nah, tulislah script-script itu di antara Sub dan End Sub itu.

Contoh yang paling sederhana adalah sebagai berikut:

```
Sub ProcedurePertamaku()
   Tampilan = "Membuat Procedure Sederhana"
   MsgBox Tampilan
End Sub
```

Kalau Anda jalankan sub procedure di atas dengan cara menekan tombol **F5**, maka akan muncul kotak dialog seperti gambar di bawah ini:

Kotak dialog yang dijalankan setelah menulis sub procedure

Pada script di atas, kita membuat sub procedure bernama: ***ProcedurePertamaku*** (setelah nama procedure selalu diakhiri dengan sepasang tanda kurung). Fungsi tanda kurung itu, apabila Anda membuat lebih dari satu sub procedure yang saling tergantung satu dengan lainnya, maka mungkin saja di dalam tanda kurung itu terdapat nama sub procedure lain.

Cara Membuat Function

Sekarang kita akan membuat function. Materi yang dikupas di bagian ini merupakan esensi dari keseluruhan buku ini sehingga Anda perlu memahami bagaimana sebuah Function dibuat.

Telah dijelaskan di atas bahwa function berkaitan dengan hitung-hitungan. Oleh karena itu, function akan sangat dekat dengan dunia matematika. Untuk membuat function baru, awali dengan menulis teks: **Function** dan nanti akan diakhiri dengan **End Function**. Contohnya sebagai berikut:

```
Function LuasPersegiEmpat(angka)
  LuasPersegiEmpat = angka * angka
End Function
```

Kita baru saja membuat function bernama: ***LuasPersegiEmpat(angka)***. Function ini hanya memiliki satu argument yang ditulis dalam tanda kurung, yaitu: ***angka*** yang nanti harus diisi oleh nilai lewat worksheet.

Ketika function ini dijalankan, maka hasil akhirnya akan berbentuk nilai baru yang diolah lewat rumus yang ada di antara Sub Function dan End Sub itu.

Untuk menjalankan function di atas, lakukan langkah-langkah sebagai berikut:

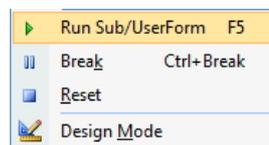
1. Pastikan Anda sudah menulis function di atas hingga tuntas.
2. Masuklah ke dalam worksheet kembali.
3. Di dalam sel A1 tulis seperti ini: ***Panjang Sisi Persegi Empat***.
4. Selanjutnya di dalam sel A2 ditulis: ***200***.
5. Tulis di dalam sel A3: ***Maka Luas Persegi Empat***.
6. Di dalam sel A4 tulis function berikut: **=LuasPersegiEmpat(A2)**.
7. Tekan tombol **Enter** dan akan diperoleh angka hasil pengkalian di dalam sel A4 itu.

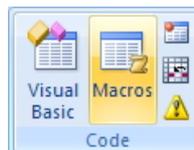
Hasil dari pembuatan function bernama "Perkalian"

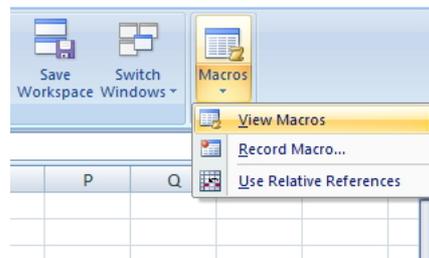
Masih ada beragam metode lainnya untuk menjalankan sebuah Function. Namun cara di atas adalah cara paling ideal jika Anda tidak puas dengan koleksi Function yang sudah disediakan oleh MS Excel.

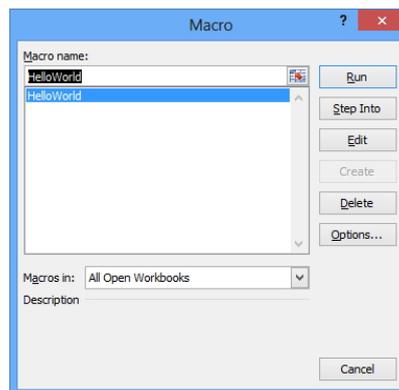
Cara Eksekusi Sub Procedure

Kalau Anda sudah membuat sub procedure, maka salah satu metode untuk menjalankan sub procedure yang sudah dibuat adalah dengan menekan tombol **F5**. Selain itu, masih ada berbagai metode untuk menjalankan Sub Procedure, antara lain:

- Jika Anda membuat beberapa sub procedure di dalam satu jendela Code yang sama, maka Anda bisa klik menu **Run** dan pilihlah **Run Sub/UserForm**.

Pilih Run Sub/UserForm untuk menjalankan sub procedure

- Kalau Anda berada di dalam worksheet, maka Anda bisa klik Tab **Developer** dan pilih ikon **Macros**.

Memilih ikon Macros yang ada di dalam Tab Developer

- Klik Tab **View** dan klik pada anak panah kecil di bawah ikon **Macros.** Pilihlah **View Macros.**

Memilih opsi View Macros yang ada di bawah ikon Macros

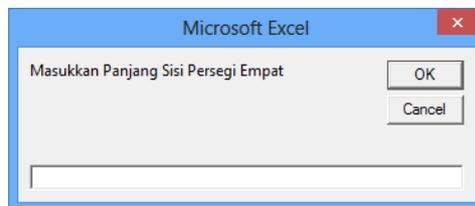
Nanti jika muncul jendela **Macro**, tekanlah tombol **Run** untuk menjalankan sub procedure itu.

Jendela Macro untuk menjalankan sub procedure yang sudah dibuat

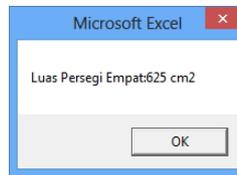
Selain berbagai metode di atas, Anda juga bisa menekan tombol shortcut yang Anda atur ketika Recording Macro. Selain itu, Macro juga bisa dipicu dengan memanfaatkan tombol atau shape di dalam worksheet, memasang ikon di dalam Quick Access Toolbar, atau item di dalam ribbon yang terasosiasi pada sebuah sub procedure.

Ikutilah langkah-langkah sebagai berikut:

1. Mulailah dengan membuat workbook baru dan aktifkan Visual Basic Editor (tekan tombol **Alt+F11** untuk mempercepat proses tersebut).
2. Klik-kanan di atas panel **Project** dan pilihlah **Insert > Module**.
3. Tulislah script sebagai berikut di dalam jendela Code:

```
Sub LuasPersegiEmpat()
  angka = InputBox("Masukkan Panjang Sisi Persegi Empat")
  MsgBox "Luas Persegi Empat:" & angka * angka & " cm2"
End Sub
```

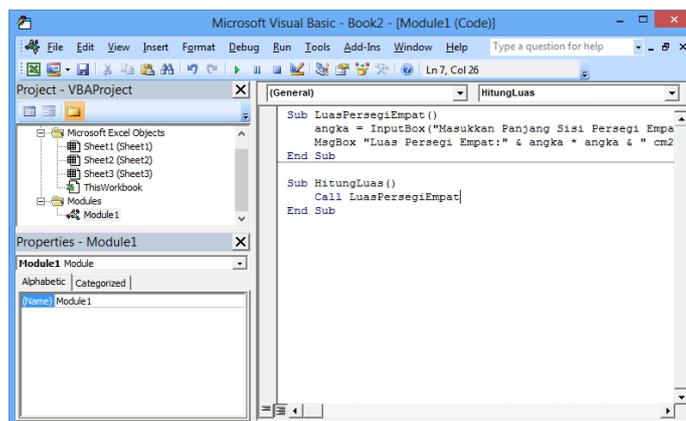
4. Jalankan sub procedure di atas lewat langkah yang paling praktis, yaitu dengan menekan tombol **F5**.

Kotak dialog akan muncul begitu Anda menekan tombol F5 untuk mengeksekusi sub procedure di atas

5. Isilah angka pada kotak teks yang telah disediakan.
6. Tekanlah tombol **OK** dan akan terlihat hasil akhir seperti gambar di bawah ini.

Hasil perkalian angka yang didapat lewat rumusan sub procedure

Mengeksekusi Macro Menggunakan Sub Procedure

Anda bisa menjalankan sebuah sub procedure dengan memanfaatkan sub procedure lainnya. Oleh karena itu, kita perlu kembali lagi ke dalam Visual Basic Editor. Lantas, lakukanlah langkah-langkah sebagai berikut:

1. Lanjutkan script VBA terakhir yang berisi hitung-hitungan luas persegi empat di atas.
2. Setelah itu, buatlah baris baru dan ketiklah contoh sub procedure di bawah ini:

```
Sub HitungLuas()
  Call LuasPersegiEmpat
End Sub
```

Membuat sub procedure kedua di bawah sub procedure sebelumnya

3. Klik sub procedure kedua tersebut, yang baru kita buat di atas.
4. Tekanlah tombol **F5**.

Sub procedure di mana kursor mouse berada akan dieksekusi bilamana Anda menekan tombol F5.

Metode Menjalankan Function

Berbeda dengan sub procedure, function tidaklah bisa dijalankan atau dieksekusi lewat cara-cara di atas. Function berbeda daripada procedure karena umumnya sebuah function melibatkan argument yang harus dihitung atau diolah. Oleh karena itu, proses menjalankan function hanya bisa dilakukan lewat salah satu dari dua metode berikut ini:

- Buatlah sub procedure dan jalankan function di dalam sub procedure.
- Bukalah worksheet dan manfaatkan Formula Bar untuk menjalankan function itu.

Berikut penjelasan tentang cara-cara menjalankan function dengan berbagai cara yang bisa ditempuh.

Menjalankan Function dengan Memanfaatkan Sub Procedure

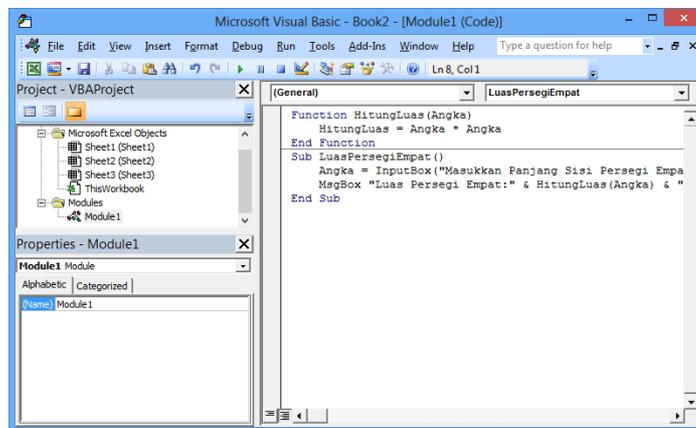
Ide dasarnya, kita "pecah" kode di atas menjadi function dan sub procedure. Jika pada contoh di atas perhitungan luas persegi empat dimasukkan langsung ke dalam sub procedure, maka pada contoh kali ini, rumus perhitungan luas persegi empat itu disendirikan ke dalam function. Lantas, hasil dari penghitungan rumus itu ditampilkan dengan menggunakan sub procedures.

Sebagai contoh, lakukan langkah-langkah sebagai berikut:

1. Tulislah script di bawah ini ke dalam jendela Code:

```
Function HitungLuas(Angka)
  HitungLuas = Angka * Angka
End Function
Sub LuasPersegiEmpat()
```

```
  Angka = InputBox("Masukkan Panjang Sisi Persegi Empat")
  MsgBox "Luas Persegi Empat:" & HitungLuas(Angka) & " cm2"
End Sub
```

2. Tekanlah tombol **F5** dan nanti akan muncul kotak dialog untuk memasukkan angka.
3. Isilah angka dan nanti angka itu akan dikalikan menggunakan rumus yang ditulis di dalam function.

Function yang ditulis di atas sebuah sub procedure yang akan mengeksekusinya

Menjalankan Function dengan Memanfaatkan Worksheet

Ini adalah cara yang paling akan sering Anda temukan nanti. Pada intinya, Anda memanfaatkan worksheet untuk menghasilkan angka atau hasil lainnya di dalam sel.

Jadi, lakukan langkah-langkah sebagai berikut untuk menjalankan function AngkaPerkalian di atas:

1. Masukkanlah sebuah angka di dalam sel, misalnya B2.
2. Klik sel yang lain dan di dalam Formula Bar, ketiklah formula: ***=HitungLuas(B2)***.
3. Tekan tombol **Enter**.

Dasar-Dasar Pemrograman VBA

Jika ingin membuat function sendiri, maka langkah paling penting yang harus dilakukan adalah memahami cara kerja *Visual Basic for Application*. Kita telah memanfaatkan secuil dari pemrograman VBA ketika membuat sub procedure dan function di atas.

Selanjutnya, Anda akan mempelajari berbagai element penting dalam pemrograman VBA, seperti comments, variable, constant, data type, arrays, dan lain sebagainya.

Menulis Komentar di dalam VBA

Para programmer sering kali membuat catatan-catatan kecil di sekitar kode-kode pemrograman untuk menandai maksud dan fungsi dari setiap baris yang mereka tulis. Catatan-catatan tersebut juga memiliki beberapa fungsi tambahan, misalnya: untuk memberi komentar atau catatan pada bagian tertentu di dalam kode pemrograman agar ketika VBA itu dilihat oleh orang lain maka tidak akan terlihat membingungkan.

Cara menyisipkan komentar di dalam script VBA tidaklah sulit. Cukup awali dengan menulis tanda petik tunggal (') di dalam jendela Code dan tulislah komentar atau catatan yang Anda kehendaki.

```
 'Fungsi berikut digunakan untuk menghitung luas persegi empat
Function HitungLuas(Angka)
   HitungLuas = Angka * Angka
End Function
```

Coba lihat script di atas. Kita membuat catatan atau keterangan untuk menjelaskan fungsi dari sebuah Function.

Bekerja dengan Variabel dan Konstanta

Ketika Anda masuk ke dalam dunia pemrograman, termasuk dalam pembuatan Function, maka tugas Anda sebenarnya adalah memanipulasi data untuk menghasilkan value baru. Dalam konteks pemrograman, data itu disimpan ke dalam "memori khusus" yang disebut variable agar lebih mudah diolah.

Dasar-Dasar Memahami Variable

Memahami variable itu sebenarnya sangatlah mudah. Langkah awal yang harus dilakukan adalah, Anda harus membuat nama untuk setiap variable yang akan Anda ciptakan. Tidak ada aturan yang terlalu spesifik tentang cara pembuatan nama itu sehingga Anda bisa membuat variable dengan nama yang informatif.

Ketika variable itu sudah diciptakan dengan cara memberinya sebuah nama, maka tugas berikutnya adalah mendeklarasikan variable itu dengan sebuah data atau nilai. Caranya hanya dengan menggunakan tanda sama-dengan (=).

Berikut contoh-contohnya:

Judul = “Membuat Fungsi dan Formula Sendiri”

Ketebalan = 150

Selain lewat metode di atas, untuk memanipulasi data Anda juga bisa mendeklarasikan value atau nilai sebuah variable dengan nilai dari variable lain. Sebagai contoh:

HalPengantar = 5

HalIsi = 150

HalDaftarPustaka = 10

HalTotal = HalPengantar + HalIsi + HalDaftarPustaka

Pada contoh di atas, variable HalTotal akan terisi oleh data angka sebesar 165.

Menentukan Tipe Data

Jika variable itu adalah “memori khusus”, atau katakanlah “ruang tempat penyimpanan”, maka itu artinya ada berbagai tipe data yang dapat disimpan ke dalam variable itu. Ada berbagai tipe data yang bisa disimpan ke dalam sebuah variable. Berikut daftarnya.

Ketika Anda membuat variable, konstanta, atau array baru, maka tugas penting yang harus dilakukan adalah menentukan tipe data yang nanti akan dideklarasikan. Di dalam dunia VBA, berbagai tipe data yang dikenali adalah sebagai berikut:

TIPE DATA	KETERANGAN
Integer	Bilangan angka dari -32.768 s.d. 32.767
Long	Bilangan angka dari -2.147.483.648 s.d. 2.147.483.648
Single	Bilangan angka negatif dari -3,402823 x 10^{38} s.d. -1,401298 x 10^{-45} Atau untuk angka positif antara 1,401298 x 10^{38} s.d. 3,402823 x 10^{38}
Double	Bilangan angka negatif dari -1,7976931348623 x 10^{308} s.d. -4,940656458623 x 10^{-324} Bilangan angka positif dari 4,940656458623 x 10^{-324} s.d. 1,7976931348623 x 10^{308}
Decimal	+/- 79228162514264337593543950335 (tanpa dilengkapi titik desimal) +/- 7,9228162514264337593543950335 (dilengkapi dengan 28 angka di belakang titik desimal)
Currency	Angka dengan nilai -922.337.203.685.477,5808 s.d. 922.337.203.685.477,5808
String	Karakter teks dari 0 sampai 2 miliar karakter
Byte	Bilangan dari 0 sampai 255
Boolean	True atau False
Date	Tanggal dari 1 Januari 100 s.d. 31 Desember 9999. Waktu dari 00:00:00 sampai 23:59:59

Object	Digunakan untuk merujuk pada objek apa pun yang diperlukan oleh script VBA agar dapat digunakan secara lebih mudah di bagian lain dalam script itu.
Variant	Teks dan angka sekaligus. Jika teks maka akan disimpan dalam bentuk String. Sedangkan kalau angka akan disimpan dalam bentuk Double.

Tiap-tiap tipe data, ketika ingin dimasukkan ke dalam variable, harus ditulis secara tepat.

Melakukan Deklarasi Variable

Setelah mengenali berbagai macam tipe data seperti tertera pada tabel di atas, maka tugas awal yang harus dilakukan saat ingin membuat variable adalah mendeklarasikan variable itu sesuai dengan tipe data yang Anda inginkan. Caranya, Anda bisa memanfaatkan pernyataan: ***Dim***.

```
Dim Konsumen As String
Dim Halaman As Integer
Dim JumlahRoyalti As Double
```

Mengenal Cakupan Variable

Variable itu bekerja dalam cakupan-cakupan tertentu yang menentukan seberapa luas variable itu bisa dipakai. Ada variable yang hanya bisa digunakan di lingkungan sub procedure/function, ada yang berfungsi secara menyeluruh dalam satu module, dan lain sebagainya. Berikut cakupan-cakupan variable:

- **Procedure only:** cakupan variable tersebut sangat terbatas, hanya bisa digunakan di dalam sub procedure/function di mana variable itu dideklarasikan.
- **Module only:** itu artinya, variable tersebut memiliki cakupan dalam satu module dan bisa berlaku dan digunakan untuk seluruh sub procedure/function yang ada di dalam module tersebut.

- **All procedures in all modules:** variable tersebut bisa digunakan dalam satu workbook sehingga bisa dipakai di dalam module-module yang saling berbeda itu. Biasa juga disebut dengan istilah **Public Variable**.

Cara Mendeklarasikan Variable Berjenis Procedure Only

Kalau Anda ingin membuat variable yang cakupannya hanya digunakan untuk satu sub procedure/function saja, maka deklarasikanlah variable itu secara langsung begitu Anda menulis Sub atau Function. Berikut contohnya:

```
Sub ProsesData()
    Dim Nama As String
    Dim Usia As Integer
    Dim HariIni As Date
' tulislah sub procedure di sini
End Sub
```

Variable itu hanya diisi data dan diolah ketika sub procedure/function dijalankan. Saat sub procedure/function itu selesai dieksekusi, maka variable tersebut dalam kondisi "kosong" lagi (baca: akan terisi data baru seperti semula).

Cara Mendeklarasikan Variable Berjenis Module Only

Karena berjenis Module Only, maka variable ini memiliki cakupan untuk seluruh sub procedure atau function yang ada di dalam satu module yang sama. Untuk mendeklarasikan variable seperti ini, Anda harus menulis variable itu pada baris pertama script VBA dan posisinya ada di luar sub procedure/function manapun.

Contohnya sebagai berikut:

```
Dim NamaKonsumen As String
Dim PerolehanRabat As Double
```

```
Sub PenghitunganRabat()
    'tulis procedure di sini
End Sub
```

Cara Mendeklarasikan Variable Berjenis Public Variable

Public Variable memiliki cakupan terluas karena bisa digunakan dalam satu workbook sehingga saat Anda membuat Function, misalnya, maka Function itu bisa digunakan dalam worksheet manapun selama masih satu workbook.

Untuk membuat public variable, cukup gunakan pernyataan berikut:

```
Public RoyaltiDasar As Double
```

Tulislah pernyataan di atas pada baris pertama ketika Anda mulai menulis script VBA dan wajib diletakkan di luar sub procedure/function manapun.

Mengenal Static Variable

Variable yang dideklarasikan dalam pemrograman Visual Basic for Application akan di-reset (dikembalikan dalam kondisi semula) saat Macro tersebut berakhir dan dijalankan kembali. Namun ada satu pengecualian ketika Anda bekerja dengan menggunakan Static Variable. Di dalam Static Variable, nilai di dalam variable itu tetap akan "bertahan" walaupun Macro telah selesai dijalankan.

Untuk membuat Static Variable, Anda bisa menulis seperti ini

```
Static UsiaHariIni As Integer
```

Jika variable itu dideklarasikan menggunakan sebuah angka dan Anda menjumlahkan angka itu menggunakan function atau sub procedure tertentu, maka angka hasil penjumlahan terakhir akan disimpan ke dalam variable tersebut. Ketika Macro itu dijalankan kembali, maka angka terakhir akan digunakan sebagai patokan yang baru.

Mengenal Konstanta

Konstanta atau constants, adalah elemen yang menyerupai variable namun value yang ada di dalamnya, begitu dideklarasikan, tidak akan bisa berubah. Jadi, ketika konstanta dibuat dan dideklarasikan dengan sebuah value, maka untuk seterusnya nilai di dalamnya akan bersifat tetap.

Untuk membuat konstanta, cukup manfaatkan statement: ***Const***. Berbeda dengan variable, pembuatan konstanta dilengkapi dengan nilai yang langsung dimasukkan ke dalam konstanta itu. Misalnya seperti ini:

```
Const Bunga As Integer = 10

Const DataNasabah As String = “Edward”
```

Karena sifatnya yang tidak bisa diubah-ubah, Anda bisa memanfaatkan konstanta untuk mereferensikan value yang pasti akan dipakai secara berulang. Contohnya: fixed rate, harga pokok penjualan, gelar, dan lain sebagainya.

Sama seperti variable, konstanta juga memiliki tiga cakupan, yaitu sub procedure/function only, module only, dan public. Cara pembuatannya tidak berbeda dengan pembuatan variable.

Bekerja dengan Menggunakan Tipe Data String

Salah satu tipe data yang bisa dideklarasikan ke dalam sebuah variable atau konstanta adalah String. Yang dimaksud String di dalam VBA adalah teks, yaitu semua karakter (huruf, angka, simbol-simbol, dan lainnya) yang ditulis di dalam tanda kutip ganda (“). Ketika Anda ingin membuat variable berjenis String, maka cukup menuliskan seperti ini:

```
Dim NamaSekolah As String
```

Selama sebuah teks ditulis dalam tanda kutip ganda, maka akan dianggap string termasuk jika Anda menulis: “12345”. Karena dianggap String, maka angka tersebut tidak bisa dijadikan patokan untuk hitung-hitungan matematika.

Bekerja dengan Menggunakan Tipe Data Date

Saat Anda membuat Function sendiri, maka Anda akan sangat sering bekerja dengan tanggal (Date). Ketika Anda bekerja menggunakan VBA, maka Anda bisa memanfaatkan tipe data Date saat ingin memanipulasi tanggal itu. Berikut cara membuat variable dengan tipe data Date:

```
Dim TanggalTransaksi As Date
```

Anda juga bisa membuat konstanta apabila ingin menentukan tanggal yang bersifat tetap dan tidak bisa diubah lagi, misalnya seperti ini:

```
Const TanggalUlangTahun As Date = #1/1/1979#
Const JamMasukKerja = #08:00:00#
```

Menulis format tanggal dan waktu memang cukup *tricky* di lingkungan MS Excel. Ketika Anda bekerja dengan Visual Basic for Application, maka sistem penanggalan yang berlaku adalah sistem internasional dengan format mm/dd/yyyy. Namun, ketika variable itu ditampilkan di layar monitor maka format tanggal yang terlihat akan mengikuti sistem yang berlaku di komputer Anda. Dengan demikian, kalau Anda mengatur sistem MS Windows dengan penanggalan dd/mm/yyyy maka tanggal tersebut akan dimulai dari tanggal-bulan-tahun.

Bekerja dengan Menggunakan Arrays

Arrays merupakan variable-variable dengan nama yang sama namun menyimpan value berbeda-beda. Untuk alasan kepraktisan, variable-variable itu cukup diberi nama sama namun dengan "penanda" yang berbeda. Sebagai contoh, Anda ingin membuat variable-variable untuk menyimpan 10 nama nasabah. Alih-alih membuat 10 variable dengan nama berbeda seperti: Nasabah1, Nasabah2, Nasabah3, … Nasabah10, maka lebih baik buatlah Arrays. Cara membuat arrays tidak jauh berbeda dengan membuat variable biasa. Hanya saja, kita menentukan jumlah arrays itu. Contohnya seperti ini:

```
Dim Nasabah (0 To 9) As Integer
```

Jumlah array dimulai dari 0 sehingga kalau menulis (0 To 9), maka ada 10 variable. Anda juga bisa menulis total jumlah variable yang ingin dibuat sehingga pada contoh di atas, Anda bisa menulis:

```
Dim Nasabah (9) As Integer
```

Ketika sudah membuat arrays, maka Anda bisa mengisi variable-variable itu satu demi satu seperti ini:

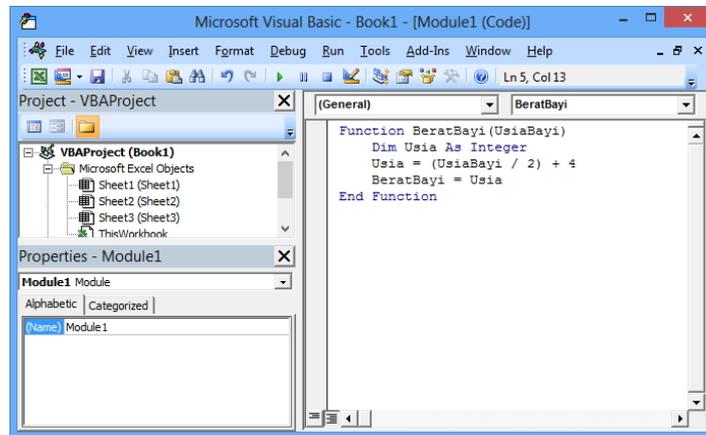
```
Nasabah(1) = “Felix Setiawan”
Nasabah(2) = “Joko Lodang”
Nasabah(3) = “Ganjar Pranowo”
```

Contoh Function yang Melibatkan Penggunaan Variable

Variable dan konstanta merupakan dua elemen yang paling mendasar dari sebuah pemrograman, termasuk saat Anda membuat Function sendiri. Berikut contoh Function yang dibuat menggunakan variable.

Untuk membuat function penghitungan bonus di atas, maka lakukan langkah-langkah sebagai berikut:

1. Buat workbook baru dan Module dari awal lagi.
2. Tulislah function seperti berikut ini:

```
Function BeratBayi(UsiaBayi)
   Dim Usia As Integer
   Usia = (UsiaBayi / 2) + 4
   BeratBayi = Usia
End Function
```

Function yang ditulis di dalam Visual Basic Editor

3. Untuk menggunakan function di atas, masuklah ke dalam worksheet MS Excel.
4. Buatlah tabel seperti gambar di bawah ini dan klik di sel E4 pada contoh tabel di bawah ini:

NAMA DAN BERAT IDEAL BAYI

No.	Nama	Usia (Bulan)	Berat Ideal
1	Joni	8	
2	Abdul	9	
3	Erik	4	
4	Bowo	5	
5	Hendrik	12	
6	Bin	24	
7	Arnold	36	

Data penjualan masing-masing sales

5. Ketik formula sebagai berikut: ***=BeratBayi(D4)***.
6. Tekan tombol **Enter** dan nanti akan muncul hasil dalam bentuk angka.
7. Klik-drag kotak sel E4 itu ke arah bawah sampai E10 sehingga masing-masing sel E4-E10 terisi oleh data bonus yang berbeda-beda (tergantung angka di kolom **Usia(Bulan)**).

E4 =BeratBayi(D4)

NAMA DAN BERAT IDEAL BAYI

No.	Nama	Usia (Bulan)	Berat Ideal
1	Joni	8	8
2	Abdul	9	8
3	Erik	4	6
4	Bowo	5	6
5	Hendrik	12	10
6	Bin	24	16
7	Arnold	36	22

Penghitungan hasil bonus lewat formula yang ditulis menggunakan VBA

Memilih Tipe Data yang Tepat

VBA memiliki banyak tipe data yang bernuansa angka. Sebut saja: Integer, Decimal, Byte, dan lain sebagainya. Nah pertanyaannya, manakah yang akan kita pakai?

Jawabannya mungkin akan sangat bervariasi, namun pedomannya sebagai berikut:

- Secara umum, Anda bisa menggunakan Integer sebagai tipe data angka yang paling aman untuk dipakai sejauh hasil kalkulasi yang didapatkan tidak berpotensi mengarah ke bentuk angka desimal. Sebagai contoh, hitung-hitungan prosentase atau bunga bank lebih baik tidak memanfaatkan Integer.
- Double adalah tipe data angka yang sangat besar. Untuk hitung-hitungan yang hanya menghasilkan beberapa digit angka, maka gunakan saja selain Double untuk menghemat resource komputer.
- Untuk penghitungan uang, maka lebih aman jika memakai tipe data Currency. Ketika berjenis Currency, maka Anda bisa menambahkan mata uang di bagian depan angka yang dihasilkan di dalam sel hanya dengan memanfaatkan pilihan **Number Format** dan memilih **Currency**.